

### **Remarks**

The above Amendments and these Remarks are in reply to the Office Action mailed August 9, 2005. A Petition for Extension of Time is submitted herewith, together with the appropriate fee.

#### **I. Summary of Examiner's Rejections**

Prior to the Office Action mailed August 9, 2005, Claims 1-18, 21 and 22 were pending in the Application. In the Office Action, Claims 17 and 18 were rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. Claims 1-18, 21 and 22 were rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,877,163 to Jones et al. (hereinafter Jones) in view of U.S. Patent No. 6,463,460 to Simonoff.

#### **II. Summary of Applicant's Amendment**

For the purpose of expediting prosecution of this application, Applicant herein adds new claims and presents some amendments that will further highlight the distinctions between claimed embodiments of the present invention and the cited references. Applicant reserves the right to pursue any earlier presented claims in a currently pending or continuation application, without prejudice to or disclaimer of the earlier presented claims.

The present Response amends the Specification in order to correct various minor informalities. No new matter has been added. The present Response also amends Claims 1, 10, 13 and 16, cancels Claims 11, 12, 14 and 15 and adds new Claims 23-25, leaving for the Examiner's present consideration Claims 1-10, 13, 16-18 and 21-25. Reconsideration of the Application, as amended, is respectfully requested. Applicant respectfully reserves the right to prosecute any originally presented or canceled claims in a continuing or future application.

#### **III. Claim Rejections under 35 U.S.C. § 112**

In the Office Action mailed August 9, 2005, Claims 17 and 18 were rejected under 35 U.S.C. 112, first paragraph as failing to comply with the written description requirement. In particular, the

Claims 17 and 18 recite the limitation "without requiring a relinking of the application and a vendor software package." Per Office Action, "there does not appear to be a written description of the claimed limitation in the specification as filed."

Applicant respectfully disagrees. Since the wrapper object is generated dynamically at run time (e.g., par. [0016]), it is at least implicit within the disclosure that the wrapper object enables the application to access wrapped vendor objects without requiring a relinking of the application and a vendor software package, as recited in Claims 17 and 18. Furthermore, because in one embodiment, the wrapper class includes all public interfaces implemented by the vendor class required by the application program, it is implicit that no relinking is required in order to allow the application access to the wrapped vendor class.

As such, Applicant respectfully submits that Claims 17 and 18 are fully supported by the Specification as filed and requests reconsideration of the rejections.

#### **IV. Claim Rejections under 35 U.S.C. §103(a)**

In the Office Action mailed August 9, 2005, Claims 1-18, 21 and 22 were rejected under 35 U.S.C. 102(b) as being unpatentable over U.S. Patent No. to Jones et al. (hereinafter Jones) in view of U.S. Patent No. 6,463,460 to Simonoff.

#### **Claim 1**

Claim 1 has been amended to more explicitly define the embodiment therein. As amended, Claim 1 now reads:

*1. A machine-readable medium carrying one or more sequences of instructions for dynamically generating a wrapper object, which instructions, when executed by one or more processors, cause the one or more processors to carry out the steps of:*  
*receiving a vendor defined class and a superclass;*  
*performing reflection on the vendor class to obtain vendor specific extension methods defined within the vendor class the reflection including retrieving meta information for allowing a server to generate a wrapper class that matches the vendor class;*

*generating the wrapper class as a subclass of the superclass, wherein the wrapper class comprises at least one of the vendor specific extension methods from the vendor class;*

*generating a wrapper object as an instance of the wrapper class by instantiating the wrapper class, thereby associating a relationship between the wrapper object and the vendor object; and*

*providing the wrapper object to an application program, thereby providing the application program with access to vendor specific extension methods.*

Thus, Claim 1 defines performing reflection on the vendor class to obtain vendor specific extension methods by retrieving meta information which allows a server to generate a wrapper class that matches the vendor class. Claim 1 also defines generating a wrapper class for the vendor class where the wrapper class comprises at least one of the vendor specific methods from the vendor class. Furthermore, Claim 1 defines generating a wrapper object from the wrapper class thereby creating a relationship between the wrapper object and the vendor object.

The Office Action mailed on August 8, 2005 agreed that Jones does not teach a wrapper class comprising at least one of vendor specific extension methods from the vendor class. The Office Action asserted that while Jones does not teach a wrapper class comprising at least one of vendor specific extension methods from the vendor class, such a feature was taught by Simonoff. According to Office Action:

*"Simonoff teaches a wrapper class comprising at least one of vendor specific extension methods from the vendor class (wrapper object provides for an open architecture design so that developing new objects for user with the White Board is greatly simplified. Stated another way, the wrapper allows third party objects to simply plug-in to the White Board) and providing the wrapper object to an application program, thereby providing the application program with access to vendor specific extension methods." (Office Action page 5).*

Applicant respectfully disagrees. Simonoff appears to merely teach a wrapper object for allowing a third party object to plug-in to the White Board system, (i.e. server and clients). (Simonoff, col. 16, lines 50-57). The wrapper object as taught in Simonoff, appears to be used for telling the

White Board system various characteristics about the object that it wraps, such as the kind of object to display, its location or size. (Col. 16, lines 44-50). This is not the same as a wrapper object which provides an application program access to vendor specific extension methods, as recited in Claim 1. The wrapper object recited in Claim 1 is used to intercept method invocations between the application and the vendor object, thereby allowing for various advantages such as performance of pre-invocation processing, post invocation processing, access to standard and non-standard vendor extensions as well as others. Thus, Simonoff also fails to disclose this feature of Claim 1.

Furthermore, Applicant respectfully submits that Jones and Simonoff also fail to disclose performing reflection on the vendor class by retrieving meta information for allowing a server to generate a wrapper class that matches the vendor class, as defined by amended Claim 1. Similarly, the references fail to disclose associating a relationship between the wrapper (proxy) object and the vendor object, as recited in Claim 1.

Jones teaches a system and method for dynamic proxy classes. In particular, Jones appears to disclose a proxy class which provides uniform access to multiple types of interfaces implemented by the proxy class. Furthermore, Jones appears to disclose an instance of that proxy class. However, Jones fails to create a relationship between that instance and the object that processes the invocation of a method. Similarly, Jones fails to teach performing reflection on the vendor class which allows a server to generate a wrapper class that matches the vendor class.

Instead, Jones teaches that a proxy instance is created via reflection on the proxy class (col. 8, lines 25-35). It fails to teach creating a wrapper class by performing reflection on the *vendor class*, thereby allowing the server to create a wrapper class that *matches* the vendor class, as defined in Claim 1. Similarly, no relationship appears to be associated in Jones between the generated wrapper (proxy) object and the vendor object. Instead, Jones appears to only define a relationship between the proxy object and the invocation handler (col. 3, lines 42-47).

Simonoff fails to fix the shortcomings of Jones in teaching the foregoing features of Claim 1. Thus, in view of the above comments, Applicant respectfully submits that Claim 1, as amended,

is neither anticipated by, nor obvious in view of the cited references, and reconsideration thereof is respectfully requested.

#### **Claim 10**

Claim 10 has been amended to more explicitly define the embodiment therein. As amended, Claim 10 now reads:

*10. A machine-readable medium carrying one or more sequences of instructions for processing an invocation at a dynamically generated wrapper, which instructions, when executed by one or more processors, cause the one or more processors to carry out the steps of:*

- receiving, from an application program, an invocation call directed to a wrapped vendor object;*
- initiating pre-processing by calling a pre-invocation handler configured to execute server-side code;*
- calling the wrapped vendor object;*
- receiving a result from the wrapped vendor object;*
- initiating post-processing by calling a post-invocation handler configured to execute post processing server-side tasks; and*
- providing the result to the application program, thereby enabling the application program to access vendor specific extension methods of the wrapped vendor object.*

As amended, Claim 10 defines initiating pre-processing by calling a pre-invocation handler configured to execute server-side code after receiving a method invocation call from an application. Similarly, Claim 10 defines initiating post-processing by calling a post-invocation handler after receiving a result from the wrapped vendor object. Applicant respectfully submits that Jones and Simonoff also fail to disclose these features.

Instead, Jones discloses an invocation handler that processes the request received from the proxy class instance. This invocation handler disclosed in Jones, is created to be application specific (col. 3, lines 42-47). Thus, it is not the same as a pre-invocation handler or a post-

invocation handlers which are configured to execute *server-side* code, such as *global* transaction processing for example.

Simonoff fails to fix the shortcomings of Jones in teaching the foregoing features of Claim 10. Thus, in view of the above comments, Applicant respectfully submits that Claim 10, as amended, is neither anticipated by, nor obvious in view of the cited references, and reconsideration thereof is respectfully requested.

#### **Claims 21 and 22**

Claims 21 and 22 are not addressed separately, but it is respectfully submitted that these claims have substantially similar features as those discussed above. Applicant respectfully submits that Claims 21 and 22 are similarly neither anticipated by, nor obvious in view of the cited references, and reconsideration thereof is respectfully requested.

#### **Claims 2-9, 13, 16-18, 23 and 25**

Claims 2-9, 13, 16-18, 23 and 25 are not addressed separately, but it is respectfully submitted that these claims are allowable as depending from an allowable independent claim, and further in view of the comments provided above. Applicant respectfully submits that Claims 2-9, 13, 16-18, 23 and 25 are similarly neither anticipated by, nor obvious in view of the cited references, and reconsideration thereof is respectfully requested.

It is also submitted that these claims also add their own limitations which render them patentable in their own right. Applicant respectfully reserves the right to argue these limitations should it become necessary in the future.

Application No.: 10/706,515  
Response to OA dated: August 9, 2005  
Response dated: December 9, 2005

### **Claims 23-25**

New Claims 23-25 have been added by this Response in order to explicitly define certain specific embodiments of the invention. Applicant respectfully requests consideration of Claims 23-25.

### **V. Conclusion**


In view of the above amendments and remarks, it is respectfully submitted that all of the claims now pending in the subject patent application should be allowable, and reconsideration thereof is respectfully requested. The Examiner is respectfully requested to telephone the undersigned if he can assist in any way in expediting issuance of a patent.

Enclosed is a PETITION FOR EXTENSION OF TIME UNDER 37 C.F.R. § 1.136 for extending the time to respond up to and including December 9, 2005.

The Commissioner is authorized to charge any underpayment or credit any overpayment to Deposit Account No. 06-1325 for any matter in connection with this response, including any fee for extension of time, which may be required.

Respectfully submitted,

Date: 12/09/2005

By:   
Justas Geringson  
Reg. No. 57,033

Customer No.: 23910  
FLIESLER MEYER LLP  
Four Embarcadero Center, Fourth Floor  
San Francisco, California 94111-4156  
Telephone: (415) 362-3800